A Proximal Minorant Method for Oracle-Structured Distributed Optimization

Tetiana Parshakova Fangzhao Zhang
Stephen Boyd
(with lots of input from others)

Stanford University

7/18/22

Outline

Oracle-structured distributed optimization

Proximal minorant algorithm

Distributed optimization overview

- distributed optimization has been studied since the 1940s
- many methods can be used for distributed optimization
  - dual decomposition
  - alternating directions method of multipliers (ADMM)
- classical setting
  - easy to evaluate agent values, (sub)gradients
  - coordinator performs simple operations, *e.g.*, averaging
- our setting
  - substantial workload per agent
  - so coordinator can do much more than, *e.g.*, just average

Oracle-structured distributed optimization problem

$$\text{minimize} \quad h(x) = f(x) + g(x)$$

- $x = (x_1, \ldots, x_M) \in \mathbf{R}^n$ is variable, $x_i \in \mathbf{R}^{n_i}$
- $f(x) = \sum_{i=1}^M f_i(x_i)$ is block separable
- f_i convex, accessed by value/subgradient oracle
- $g : \mathbf{R}^n \to \mathbf{R} \cup \{\infty\}$ is convex structured objective function
- infinite values of g encode constraints
- coordinator can solve an optimization problem involving g

Our goals

we are interested in methods that

- find good points in tens of iterations or fewer
- solve convex problems exactly
- do something reasonable when f_i aren't convex
- have *zero* hyper-parameters to tune
- handle agent delays and failures

Methods

- (accelerated) proximal subgradient
- ADMM, Douglas-Rachford
- cutting plane/bundle methods

- we've settled on cutting-plane/bundle methods (for now)
- these methods build up a piecewise linear model of each f_i

Example: Consensus problem

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^{M} f_i(x_i) \\ \text{subject to} & x_1 = \cdots = x_M \end{array}$$

- x_i are variables, f_i are convex
- to put in oracle-structured form, use

$$g(x) = \begin{cases} 0 & x_1 = \cdots = x_M \\ \infty & \text{otherwise} \end{cases}$$

(indicator function of consensus)

Agents

- when queried by coordinator at x_i, agent returns

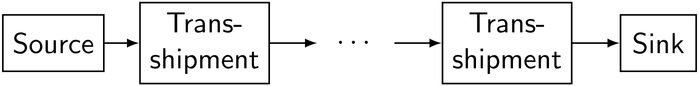
$$f_i(x_i), \qquad q_i \in \partial f_i(x_i)$$

- agents can include *private variables* z_i, with

$$f_i(x_i) = \min_{z_i} F_i(x_i, z_i)$$

- to evaluate $f_i(x_i)$ and $q_i \in \partial f_i(x_i)$ we solve an optimization problem

Example: Supply chain



- single commodity network with M trans-shipment components in series
- trans-shipment component i routes input flows $a_i \in \mathbf{R}_+^{m_i}$ to output flows $b_i \in \mathbf{R}_+^{n_i}$, with cost $f_i(a_i, b_i)$
- flow is conserved: $\mathbf{1}^T a_i = \mathbf{1}^T b_i$
- series connection means $b_1 = a_2, \ldots, b_{M-1} = a_M$
- source and sink costs $\psi^{\mathrm{src}}(a_1) + \psi^{\mathrm{sink}}(b_M)$

Oracle-structured form

- $x_i = (a_i, b_i)$
- $f_i(x_i)$ is minimum cost of trans-shipment problem with quadratic edge cost, edge capacities
- solve trans-shipment problem with $n_i m_i$ private variables to evaluate $f_i(x_i)$, $q_i \in \partial f_i(x_i)$
- structured objective term is

$$g(x) = \psi^{\text{src}}((x_1)_{1:m_1}) + \psi^{\text{sink}}((x_M)_{m_M+1:m_N}) + \mathcal{I}(x)$$

  $\mathcal{I}(x)$ is indicator function of flow constraints
- roughly speaking:
  - $f_1 + \cdots + f_M$ is the shipping cost
  - g is the negative gross profit

Outline

Oracle-structured distributed optimization

Proximal minorant algorithm

Agent objective minorants

- algorithm maintains a *minorant* $\hat{f}_i$: $\hat{f}_i(x) \leq f_i(x)$ for all x
- at iteration k, query each agent i at $x_i^{(k+1)}$ to get

$$f_i(x_i^{(k+1)}), \qquad q_i^{(k+1)} \in \partial f_i(x^{(k+1)}$$

- update minorant of agent i

$$\hat{f}_i^{(k+1)}(x_i) = \max\left(\hat{f}_i^{(k)}(x_i),\ f_i(x_i^{(k+1)}) + (q_i^{(k+1)})^T(x_i - x_i^{(k+1)})\right)$$

- update minorant of h

$$\hat{h}^{(k+1)}(x) = \hat{f}_1^{(k+1)}(x) + \cdots + \hat{f}_M^{(k+1)}(x_M) + g(x)$$

Bounds on optimal value

- $L^{(k)} = \min_x \hat{h}^{(k)}(x)$ is lower bound on optimal value
- $U^{(k)} = \min\{U^{(k-1)}, h(x^{(k)})\}$ is upper bound on optimal value
- *duality gap* is $U^{(k)} - L^{(k)}$
- *relative duality gap* is

$$\omega^{(k)} = \frac{U^{(k)} - L^{(k)}}{\min\{|U^{(k)}|, |L^{(k)}|\}}$$

- stopping criterion is $\omega^{(k)} \le \epsilon^{\text{rel}}$
- guarantees relative suboptimality less than ϵ^{rel}

Proximal minorant algorithm

Algorithm

given $x^{(0)} \in \textbf{dom}\, h$, $h(x^{(0)})$, initial minorants $\hat{f}_i^{(0)}$ and stepsize $\rho^{(0)}$.

for $k = 0, 1, \ldots$

1. *Check stopping criterion.*
2. *Update iterate.* $x^{(k+1)} = \operatorname{argmin}_x \left(\hat{h}^{(k)}(x) + (\rho^{(k)}/2)\|x - x^{(k)}\|_2^2 \right)$.
3. *Query agents.* Evaluate $f_i(x_i^{(k+1)})$ and $q_i^{(k+1)} \in \partial f_i(x_i^{(k+1)})$.
4. *Update minorants.* Update $\hat{f}_i^{(k+1)}$, $i = 1, \ldots, M$, and $\hat{h}^{(k+1)}$.

- several choices for inverse step sizes $\rho^{(k)} > 0$

Supply chain instance

- $M = 5$ trans-shipment components, with (m_i, n_i)

$$(20, 30), \quad (30, 40), \quad (40, 25), \quad (25, 35), \quad (35, 20)$$

- 300 variables; 4975 private variables
- proximal minorant algorithm $\epsilon^{\text{rel}} = 0.01$

Optimality gap (relative)